Simulation-based Evaluation of the Topic-Specific Trust Open Rating System

Holger Lewen

Institute AIFB, Universität Karlsruhe (TH), Germany
lewen@aifb.uni-karlsruhe.de

Abstract. In order to understand how algorithms work in different scenarios, it is important to test them both on real and on synthetic data. This technical report provides the description of the simulation-based evaluation of the Topic-Specfic Open Rating System. It includes a complete description of the experiment setup, a motivation and also a detailed analysis of the results.

1 Introduction

One way of evaluating the behavior of a system is to run a simulation. Using synthetic data to test the system behavior has several advantages over purely analyzing real world data. Whilst it is important to see whether the algorithms in the system hold for real world data as well, it is difficult to check the behavior in edge cases. Also there is no control over user behavior in real world test data. Therefore we decided to run a simulation for our Topic-Specific Open Rating System (TS-ORS) in addition to performance tests and a user study. The main purpose of the simulation is to check the system behavior in a controlled environment and draw conclusions for its use in real world systems. In order to achieve this, we have run the same task (computing overall ratings for ontologies based on trust in the system) several times, altering parameters in order to see how different coverage or errors made by the users affects the outcome. In the remainder of this report, the system we use as a testbed is introduced, as well as our evaluation setup. Later the results of the different test runs are shown and analyzed. We finish with a conclusion.

2 Setup

This section will provide a quick overview of the system employed to run the evaluations and the setup we created.

2.1 System

Because the performance evaluation [1] and user evaluation [2] were run on the code that was integrated into Cupboard [3], we have used the exact same codebase (of the topic-specific open rating system (TS-ORS)) for our simulation.

Because of the specific application setting, we only tested topic-specific trust for ontology properties, not for domains (since at the current moment ontology domains are not used in the system). Nevertheless, the results shown in this report based on property-specific trust can be generalized for topic-specific trust covering properties and domains. As is described in [4, 5], the main feature of the TS-ORS is to rank reviews and based on the reviews also the ontologies according to computed ratings for the ontology derived from reviews and trust information available to the system. Based on the information on user-to-user trust, the TS-ORS computes both global (not user specific) and local (user specific) trust rankings. The global trust information can be used for the users not connected to the web of trust, or not identifiable (not logged in). The web of trust (WOT) can be seen as a network where users are nodes and edges are trust statements. Global trust is not user-specific, unlike local trust, which features personalized trust information on the other users. Based on knowing who the most trusted user is (note that this can be user-specific), the system can use the star rating of the review of that user to compute an overall rating of the ontologies (again based on parameters the users can define). The main purpose of the evaluation described in this report is to find out, whether the algorithms work as intended, given different scenarios that could be encountered as such in real world systems.

2.2 Basic Experiment Setup

While it would have been possible to run the experiment with a very low number of users and ontologies, we chose to have a larger number of instances and compute the average over the results, since we felt this could even out too random results. The data we generated is on the one hand the user-base, then the ontologies and reviews for the ontologies, and ultimately the trust a user places in another user.

Users The simulated users in our case are divided into three groups, which we call *good users*, *controversial users* and *bad users*. The good users do not try to game the system (by promoting SPAM or bad ontologies), they accurately review good ontologies and bad ontologies. The controversial users represent a subgroup that is not trying to game the system but has a taste (e.g. special needs) different from the mainstream. They like controversial ontologies more than the good ones, but also do not like the bad ones. The bad users are the ones trying to game the system, they rate bad ontologies highly and try to form subgroups to boost the popularity of their infiltrated bad content.

The rationale behind choosing this setting was to mirror situations that occur in real life. These systems have a large number of users that use the system as intended, but not all users have the same taste or needs. In order to mimic the "taste" difference, we have the two groups good users and controversial users. In order to mimic malicious users, (like SPAM-bots) we have the group bad users.

For the experiments, we have created 100 users total, 60% of them are good users, 20% controversial users and 20% bad users. It has to be noted that this decision does influence the outcome, since for some measures, the largest group will automatically dominate the rankings. This does not hold true for the local trust measure. We will mention this problem several times in the remainder of the document. We chose this distribution, because we felt that in a normal, well maintained system most of the users would fall in our good users group. However, it is not important for the conclusion how the distribution was made, since the findings can be generalized for different distributions.

Ontologies With regard to the ontologies, we distinguish good ontologies, controversial ontologies and bad ontologies. The good ontologies represent overall good quality; they are also liked by the controversial users, however not as much as the controversial ontologies. The bad ontologies are of bad quality and are only promoted by bad users. Each ontology has 5 properties which are rated.

We have created 50 ontologies of which 50% were good, 20% controversial and 30% bad. The distribution of good, controversial and bad ontologies does not influence the outcome of the experiment too much, since all the trust computations were made independently. It is however true, that for the small coverage cases where we randomly chose which ontology to review, the data density in terms of ratings for the ontologies would be bigger in a larger group.

Reviews For each of the different user groups, we have defined rules, according to which the reviews are generated for each ontology and property:

- "Good" users like good ontologies, and rate them highly (5 star). They differ over the controversial ontologies, 50% like them (4 star), 50% do not like them (2 star). They all dislike bad ontologies and rate them 1 star.
- "Controversial" users like the good ontologies (4 star), but like the controversial ontologies better (5 star). The also dislike the bad ontologies (1 star).
- "Bad" users dislike (1 star) all ontologies except for the bad ontologies (5 star).

Trust In terms of trust, we have decided that each user trusts only peers in his group and distrusts the rest, that means a good user trusts all good users, but distrusts controversial users and bad users. Bad users only trust bad users, reflecting the attempt to build a network to boost their importance.

2.3 Variations During Test-Runs and Alteration of the Setup

We have created three different scenarios, each consisting of 12 test runs, resulting in 36 runs total.

Scenarios

- The first scenario is reflecting a setting similar to what Guha et al [6] used for their experiment. Trust is assigned globally, not topic-specific (that means, that it is only possible to address an overall trust in another user, not trust in certain abilities). In this setup, we have also constructed the dataset under the assumption that all users were able to rate the 5 different properties of the ontology equally well. Therefore, ratings and trust are assigned as described in the section above.
- The second scenario tries to discover what happens if users have different reviewing specialties, i.e. are only experts in one area. To mimic that, we have distinguished their reviews in good reviews, i.e. reviews in their area of expertise, and bad reviews, i.e. reviews in the area where they are not experts. Specifically, if a user has for example expertise for rating property 1, these reviews would be good and according to the schema outlined in section 2.2 above. For all other properties, the review would be marked bad and the original rating was inverted (i.e. 5 became 1, 2 became 4 and 3 stayed 3). For the bad users this distinction was not made, since the assumption was that they are not interested in providing good reviews, but just promoting their bad ontologies. Therefore their rating would remain the same (1 star for every ontology except for bad ontologies, which are rated 5 stars). Trust, however, is still assigned globally within the group, i.e. the good users trust the good users, but only 20% of the trusted reviews are actually good. In the system described by Guha et al [7, 6], only global trust is allowed.
- The third scenario has the same review setting as the second scenario (i.e. expertise only in certain areas), but allows for topic-specific trust. Here, the users are only trusted by their peers for the properties for which they provide good reviews, and not for the other ones. The exception is once again the bad user group, in which still all users trust each other globally. The idea is to test how the user can benefit from the ability to assign topic-specific trust against having to use global trust.

Sparsity of Data One of the problems in real-world systems also encountered by many recommender systems is data sparsity [8]. If only a few users review ontologies and state their trust towards other users, the algorithms have to function as well. In order to see how the algorithms can handle data sparsity, we have run each setting with a 100%, 50%, 10% and 5% coverage. That means that for the 100% case each user reviews all properties of all ontologies and states his trust for all other users. In the 10% setting each user reviews a randomly selected subset of ontologies, exactly 10% of all ontologies in the system. Also each user now only states his trust towards 10% of the other users (again, the group of users was selected randomly). For simplicity, if an ontology was part of the randomly chosen subset, all properties were reviewed by the user. So in the most sparse setting (5%), given our experiment size of 100 users and 50 ontologies, each user reviews around 2 ontologies and assigns trust towards 5 other users.

Errors in Judgement Another important aspect is that users do not always act 100% as expected, but can make mistakes (like not identifying a good user or trusting a bad user. In order to measure what effect this has on stating trust statements, which in the end are used for the computation of the trust matrices, and thus the entire ranking process, we have run each of the sparsity settings once without error, once with 10% error and once with 20% error. An error would mean that the opposite of the normal action is performed, for example a good user would trust a bad user or might distrust a good user. A 10% error would mean that out of all the trust statements given by a user, 10% are wrong, i.e. complimentary to the strategy defined in section 2.2. This holds for all user groups, it is assumed that also a bad user might make mistakes.

2.4 Remarks on Implementation

Since we have run the tests on the identical code-base as used for the Cupboard system, the trust computation was executed for each ontology–property combination separately. The global trust statements were modeled as meta-trust statements, i.e. statements of trust or distrust between two users, which are then broken down to individual trust statements on the ontology–property level by the system. If, for example, user A trusts user B globally, the system searches for all reviews by user B and assigns a trust statement to them. For the third scenario, a property-based meta-trust statement was used, i.e. the trust is only propagated to reviews for that property.

2.5 Result Generation

As mentioned before, the 3 scenarios with 4 sparsity settings and 3 error settings yielded in 36 test runs. For each run, we computed the global and local trust for each ontology–property combination and then retrieved several ratings. For each of the hereafter mentioned runs, we retrieved the rating for each ontology in the system and averaged it with the results of the other ontologies of that category. Since the computations require some parameters, we have kept these stable throughout the whole experiment. They are $\alpha = 0.7$ (used to combine TrustRank and DistrustRank) and $\nu = 0.8$ (used to weigh the top N reviews using descending importance, see equation 1). The 5 properties were weighted evenly. In the case of local trust (i.e. trust that is user-specific), we have received the rating for each user and averaged it over all users in the group. In the results we distinguish the three different ontology types good, controversial and bad.

$$w_i = \frac{(\nu)^i}{\sum_{i=1}^{N} \nu^i} \text{ with } w_i \text{ weight for } i\text{-th review, } 0 \leq \nu \leq 1 \tag{1}$$

- First of all, we computed the average rating. This metric is the easiest metric obtainable and does not require any trust computation in the background.

- Then we computed the average rating based on all reviews and global trust (this means, that also bad reviews will be taken into account, but according to their position in the result set, the impact is minimal because of the decrease of importance according to position (see Equation 1).
- After that we compute the average rating based on the top 3 reviews and global trust (this means that only the first 3 reviews according to global trust measures will be taken into account).
- Finally we compute all the local trust combinations, i.e. for each of the three ontology types, we measure the average rating for each of the three user types, yielding in 9 combinations. We compute this measure once based on the top 3 reviews, and once based on the top review, all based on local trust for each individual user in the group. That means that for the case of good ontologies and good users, we retrieve for each of the 60 good users the rating for each of the 25 good ontologies. Then we compute the average.

3 Results

In this section we will discuss and analyze the outcome of the simulation.

3.1 Expected Results

Since we have clearly defined rules how the three user groups act, we would imagine the rating algorithms to also retrieve ratings according to these rules in the case of local trust. That means that if all controversial users rate controversial ontologies 5 star, this is supposed to be the rating returned by the system for controversial ontologies and controversial users when only users in the same group are trusted. It is clear that for the global measures the size of the groups matters, and the question of whether the groups are confined or whether there are links between the groups. That means that in our case, since the good users group is the largest (60%), and there are no links between the groups, we expect their opinion to dominate the ratings based on global trust. For the local trust, it is important that the trust is accurately assigned. If for example a bad user is trusted by a good user, this will also connect the good user to the bad reviews, and will influence the computed rating for the ontologies. We expect to see this effect when introduce the element of chance to simulate erroneous behavior. We furthermore assume the results to be more accurate the more trust statements and rating coverage we have. This is due to the fact that the local trust based ranking has to rely on global trust values for the users, for which no trust information can be computed by means of local trust (i.e. they are not connected in the WOT). If a bad users is not connected to other bad users for example, the system has to refer to global trust ranking of users, leading to a result different from what was expected. We furthermore expect the topic-specific trust approach to be more accurate in the case that reviewers have different reviewing skills for different properties of the ontologies (comparison between scenario 2 and 3). Lastly, we will analyze the effect of data sparsity on the rating results in section 10.

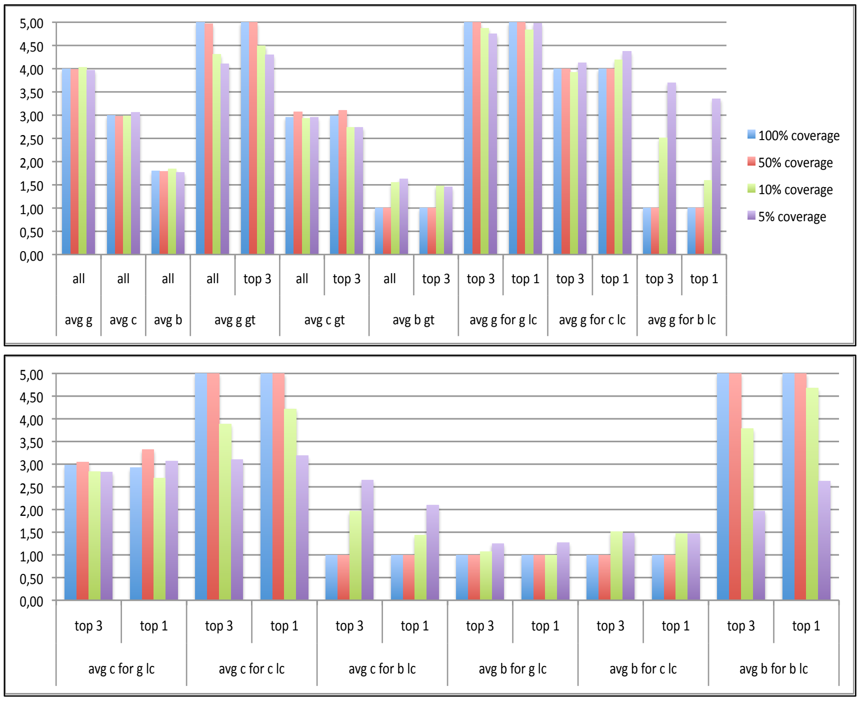
3.2 Scenario 1

As mentioned before, each scenario consists of 12 runs total, 4 different data density settings, and 3 different error settings. In the figures you will see the results for the different density settings side-by-side in one diagram. We have one figure for each error setting.

0% Error The 0% error setting is the one that is supposed to work as described in section 3.1. If we look at the results shown in Figure 1, it is evident that given perfect coverage, the algorithms indeed produce the results expected. The average rating for controversial ontologies and controversial users is 5 stars, the same holds true for good users and good ontologies. Furthermore, the bad ontologies are rated 1 star for both good and controversial ontologies. As coverage goes down, the results move towards majority taste. The is due to the size of the test data we have used and will be explained further in section 3.5. Basically the problem is that a user is not always connected to the WOT and when he is not connected, the global trust metric has to be consulted for trust information. In these sparse density cases the top1 based rating produces better results than the top3 setting. This is due to the fact that if there are less than 3 locally trusted reviews, globally trusted reviews are factored into the equation. As a general observation, global trust outperforms the simple average and local trust outperforms global trust (with top1 outperforming top3), with outperforming defined as being closer to our expected results.

10% Error The results for our 10% error setting can be found in Figure 2. One main observation is that once the boundaries of the peer group are violated by trusting a user from another peer group and furthermore also distrusting some of your peers, the largest group becomes predominant. The good users group is the biggest one in our experiment, therefore dominating the results of the controversial and bad users. Unlike the setting with no errors, these groups cannot retrieve a review from their peers as top review, since once one user form the group trusts another user form the good user group, this user will receive enough trust to become the predominant reviewer. So in this setting the expected results of 5 for controversial users and controversial ontologies is not encountered, neither the rating of 5 for bad ontologies and bad users. The difference in coverage can be explained as in the other scenario without error, namely a lack of connectedness to the WOT.

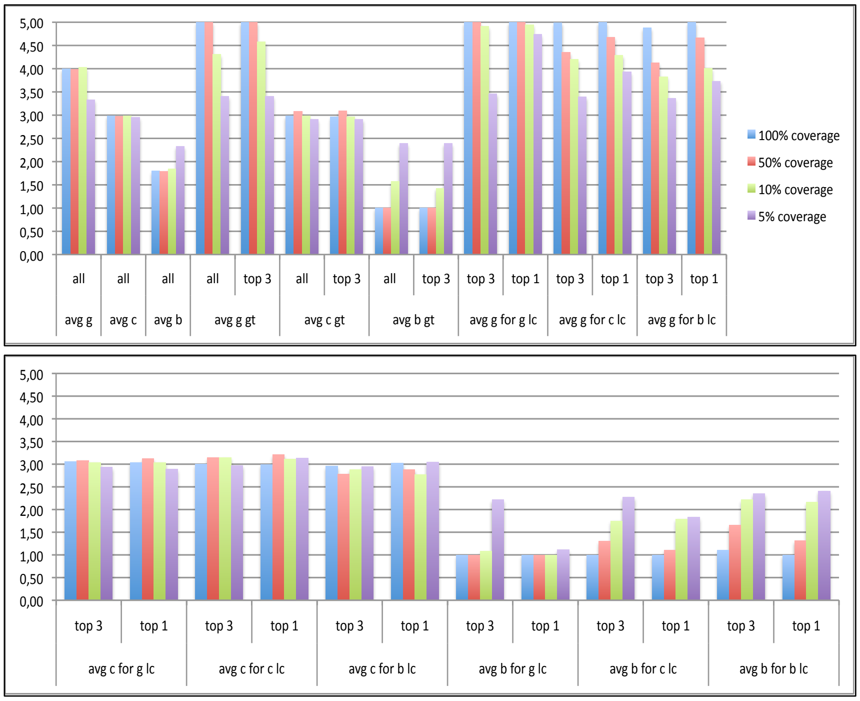
20% Error In a setting with 20% error (see Figure 3), the results resemble very much the results of the 10% setting. The effects described above are just more visible. This also allows for the conclusion that within a certain range of error, the influence on the result is minimal. That means that there is not much difference between making errors in 10% of the cases or or 20%.



Abbreviations:

avg = average
g = good (ontology / user)
c = controversial (ontology / user)
b = bad (ontology / user)
all = based on all reviews
top 3 = based on top 3 reviews
top 1 = based on top review
gc = global trust (trust that is user independent)
lc = local trust (trust that is user dependent)
avg x for y lc = average ratings of x (e.g. good) ontologies for y (e.g. controversial) users based on localtrust

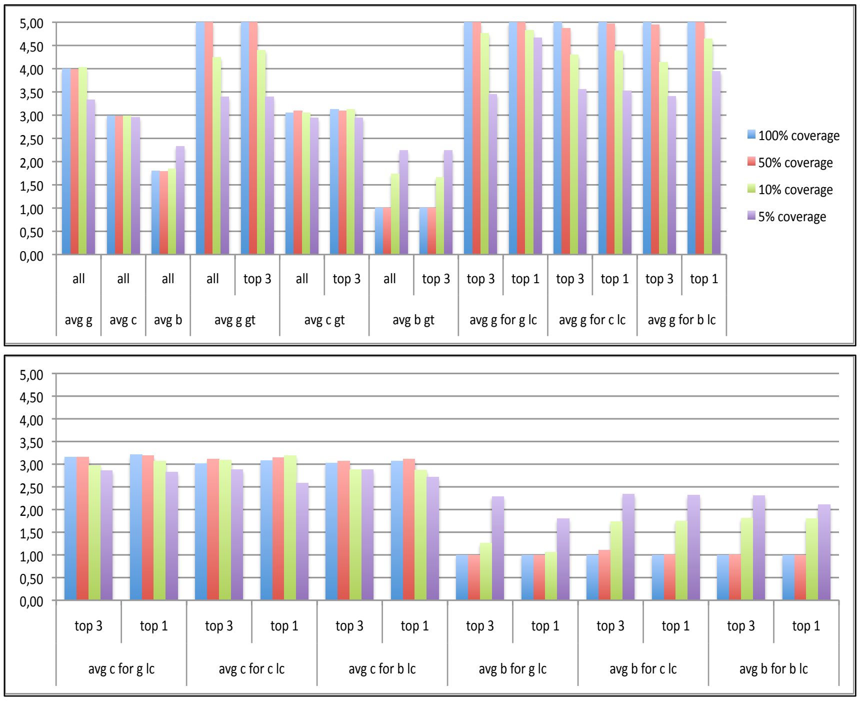
Fig. 1. First Scenario with 0% Error



Abbreviations:

avg = average
g = good (ontology / user)
c = controversial (ontology / user)
b = bad (ontology / user)
all = based on all reviews
top 3 = based on top 3 reviews
top 1 = based on top review
gc = global trust (trust that is user independent)
lc = local trust (trust that is user dependent)
avg x for y lc = average ratings of x (e.g. good) ontologies for y (e.g. controversial) users based on localtrust

Fig. 2. First Scenario with 10% Error



Abbreviations:

avg = average
g = good (ontology / user)
c = controversial (ontology / user)
b = bad (ontology / user)
all = based on all reviews
top 3 = based on top 3 reviews
top 1 = based on top review
gc = global trust (trust that is user independent)
lc = local trust (trust that is user dependent)
avg x for y lc = average ratings of x (e.g. good) ontologies for y (e.g. controversial) users based on localtrust

Fig. 3. First Scenario with 20% Error

3.3 Scenario 2

To see which drawbacks allowing only global trust statements has on the ranking results, in the second scenario we assume that each user only has one area of expertise (in our setting one ontology property), which he can review properly. The rest is also reviewed, but the reviews are of bad quality (to mirror that, we inverted the correct rating for the ontology-user combination. In order to at least get good results for the properties that these users can review, they are trusted globally (remember, that there is no more fine-grained way to assign trust in this scenario). The alternative of not trusting the user at all (because after all 80% of the results are bad) is not an alternative, since this way nobody could be trusted and the rating would have to completely rely on global trust, which would then favor the bad users who trust each other.

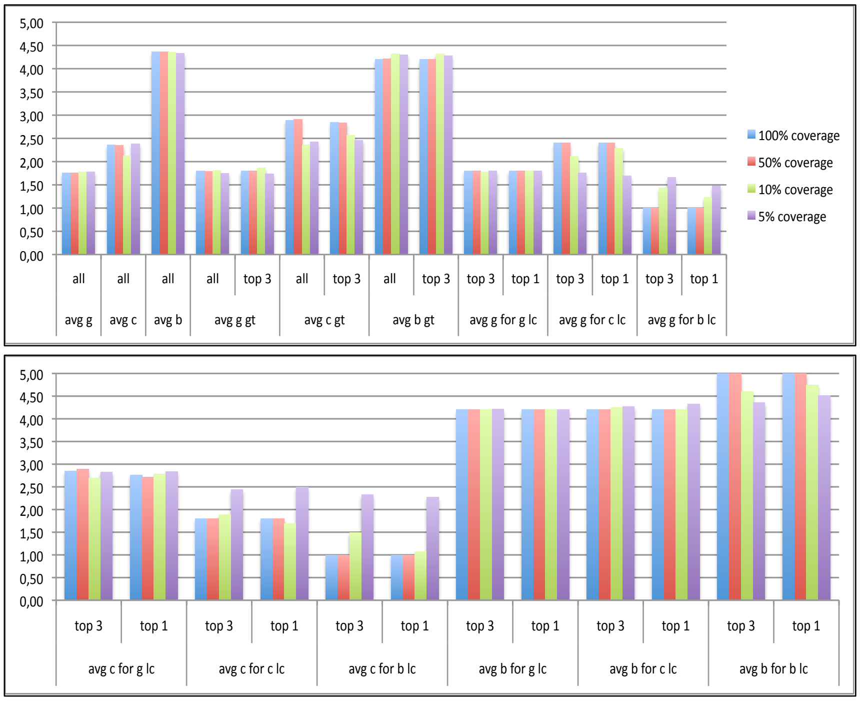
0% Error In this setting we expect to see incorrect ratings to be computed due to the fact that the global trust statements assigned based on the good reviews also cover all the bad reviews of that user. As can be seen in Figure 4, the results are the complete opposite of what would be desired. The good ontologies rank lowest and the bad ontologies first. So the expectations were fulfilled, and the fact that in the end 80% of the reviews were incorrect (only 1 out of 5 properties could be reviewed correctly) led to the expected bad results. Still, the same behavior as described in scenario 1 is encountered here. The lower the coverage, the worse the connectedness to the WOT, and so the average rating for bad ontologies for bad users drops from 5 to roundabout 4 (now dominated by the rating of the good users. Please remember that the bad users were still expected to review maliciously for all ontologies, which explains the 5 star given sufficient coverage.

10% and 20% Error Here the same observations can be made as in scenario 1, i.e. the predominant user group affects the ratings of the other subgroups. The results can be found in figures 5 and 6.

3.4 Scenario 3

Since the claim is that topic-specific trust can overcome the drawbacks of global trust, i.e. having to trust users also for bad reviews when trusting the good ones, we expect that the outcomes will be similar to scenario 1. In this setup the users are only trusted for the properties for which they can provide good reviews, and not distrusted otherwise.

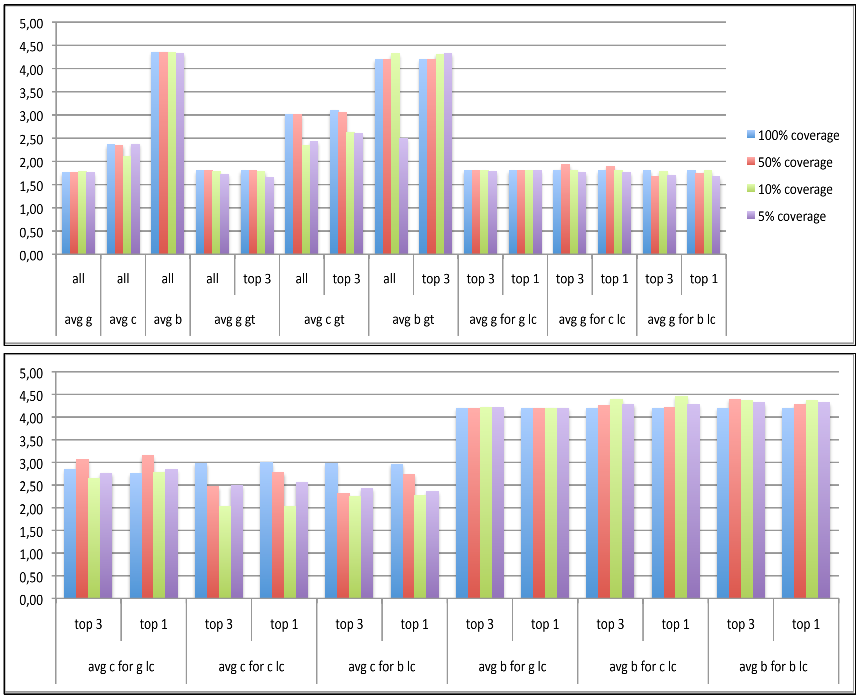
0% Error As is evident from Figure 7, the topic-specific trust indeed does provide the expected ratings given enough connectivity to the WOT. Even though the user reviews are the same as in scenario 2 (as can be seen from the overall average ratings), the ratings for local trust-based and global trust-based results are as good as in scenario 1. So even though there are 80% wrong reviews in



Abbreviations:

avg = average
g = good (ontology / user)
c = controversial (ontology / user)
b = bad (ontology / user)
all = based on all reviews
top 3 = based on top 3 reviews
top 1 = based on top review
gc = global trust (trust that is user independent)
lc = local trust (trust that is user dependent)
avg x for y lc = average ratings of x (e.g. good) ontologies for y (e.g. controversial) users based on localtrust

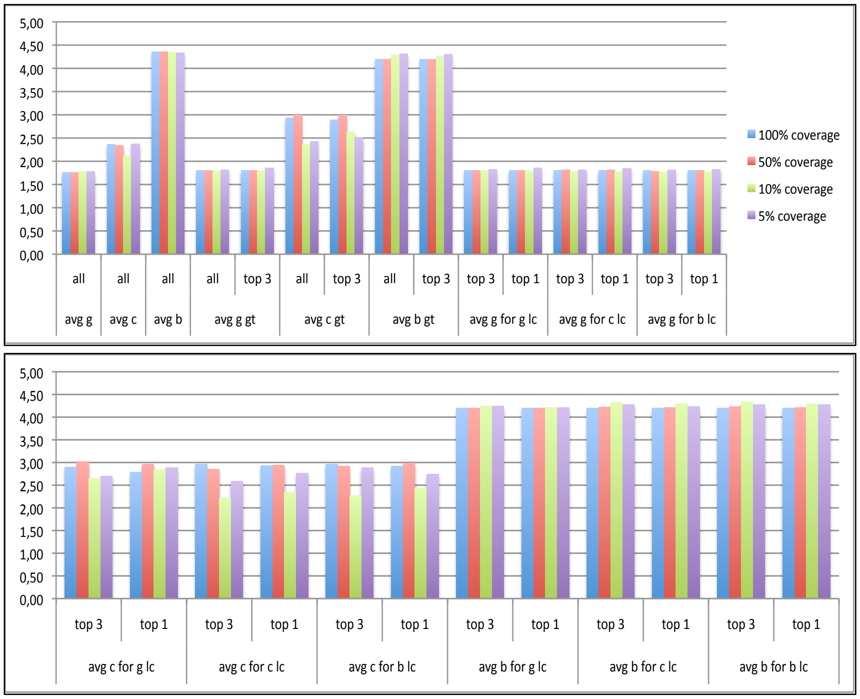
Fig. 4. Second Scenario with 0% Error



Abbreviations:

avg = average
g = good (ontology / user)
c = controversial (ontology / user)
b = bad (ontology / user)
all = based on all reviews
top 3 = based on top 3 reviews
top 1 = based on top review
gc = global trust (trust that is user independent)
lc = local trust (trust that is user dependent)
avg x for y lc = average ratings of x (e.g. good) ontologies for y (e.g. controversial) users based on localtrust

Fig. 5. Second Scenario with 10% Error

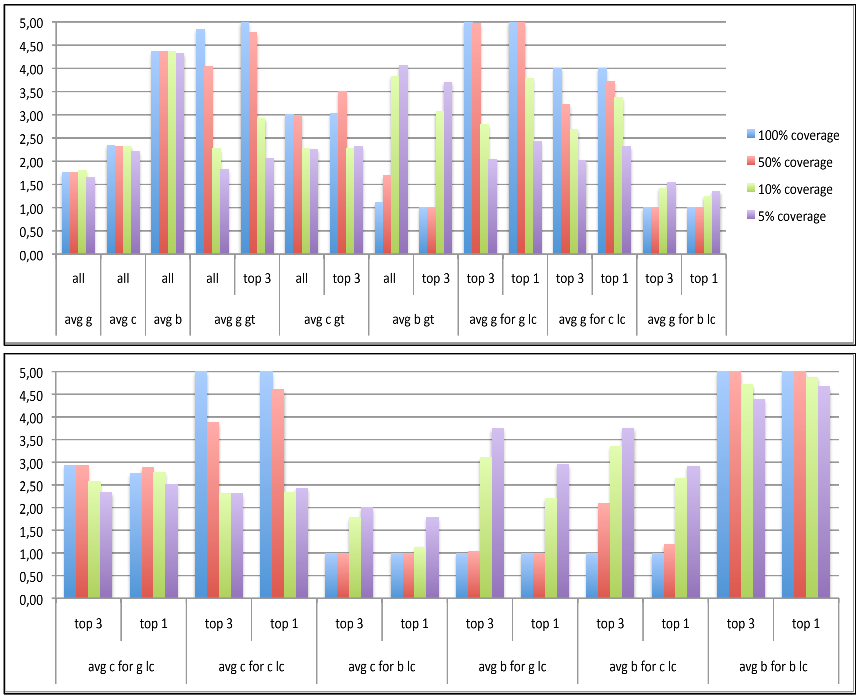


Abbreviations:

avg = average
g = good (ontology / user)
c = controversial (ontology / user)
b = bad (ontology / user)
all = based on all reviews
top 3 = based on top 3 reviews
top 1 = based on top review
gc = global trust (trust that is user independent)
lc = local trust (trust that is user dependent)
avg x for y lc = average ratings of x (e.g. good) ontologies for y (e.g. controversial) users based on localtrust

Fig. 6. Second Scenario with 20% Error

the system, the algorithms still produce the desired outcome, based on accurate trust assignment and sufficient connectivity. With decreasing coverage we can once again see how the expected results blend in with the global trust-based results for the respective coverage.

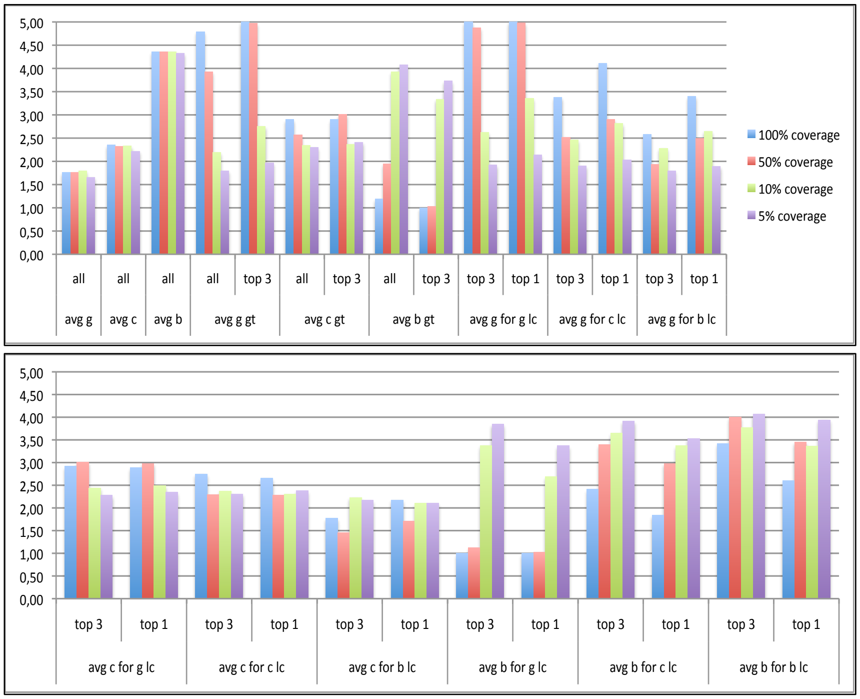


Abbreviations:

avg = average
g = good (ontology / user)
c = controversial (ontology / user)
b = bad (ontology / user)
all = based on all reviews
top 3 = based on top 3 reviews
top 1 = based on top review
gc = global trust (trust that is user independent)
lc = local trust (trust that is user dependent)
avg x for y lc = average ratings of x (e.g. good) ontologies for y (e.g. controversial) users based on localtrust

Fig. 7. Third Scenario with 0% Error

10% and 20% Error Also in scenario 3 we can find the described effect of the predominant group influencing the rating results of the other subgroups (see figures 8 and 9. Still it is noteworthy that even with 20% of all trust statements being wrong, the average rating is outperformed. In other words, even if not all trust information is correct, it is better to employ trust-based algorithms for ranking than relying on basic arithmetic measures like average.



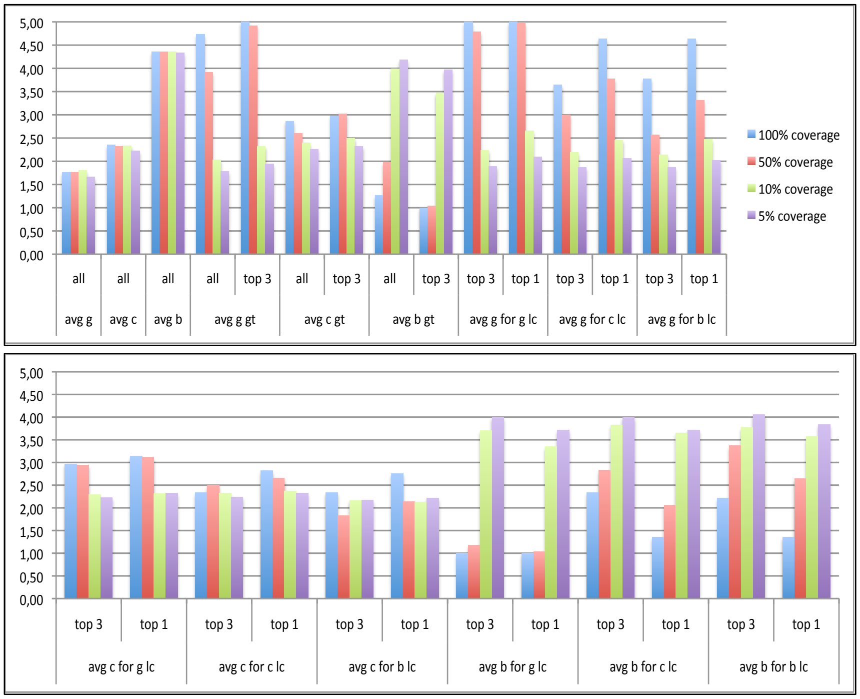
Abbreviations:

avg = average
g = good (ontology / user)
c = controversial (ontology / user)
b = bad (ontology / user)
all = based on all reviews
top 3 = based on top 3 reviews
top 1 = based on top review
gc = global trust (trust that is user independent)
lc = local trust (trust that is user dependent)
avg x for y lc = average ratings of x (e.g. good) ontologies for y (e.g. controversial) users based on localtrust

Fig. 8. Third Scenario with 10% Error

3.5 Comparison 5% Coverage with 100 and 1000 Users, 0% Error

As indicated before, we expected the worse performance of the algorithms at smaller coverage levels to be due to the missing connectedness of users to the



Abbreviations:

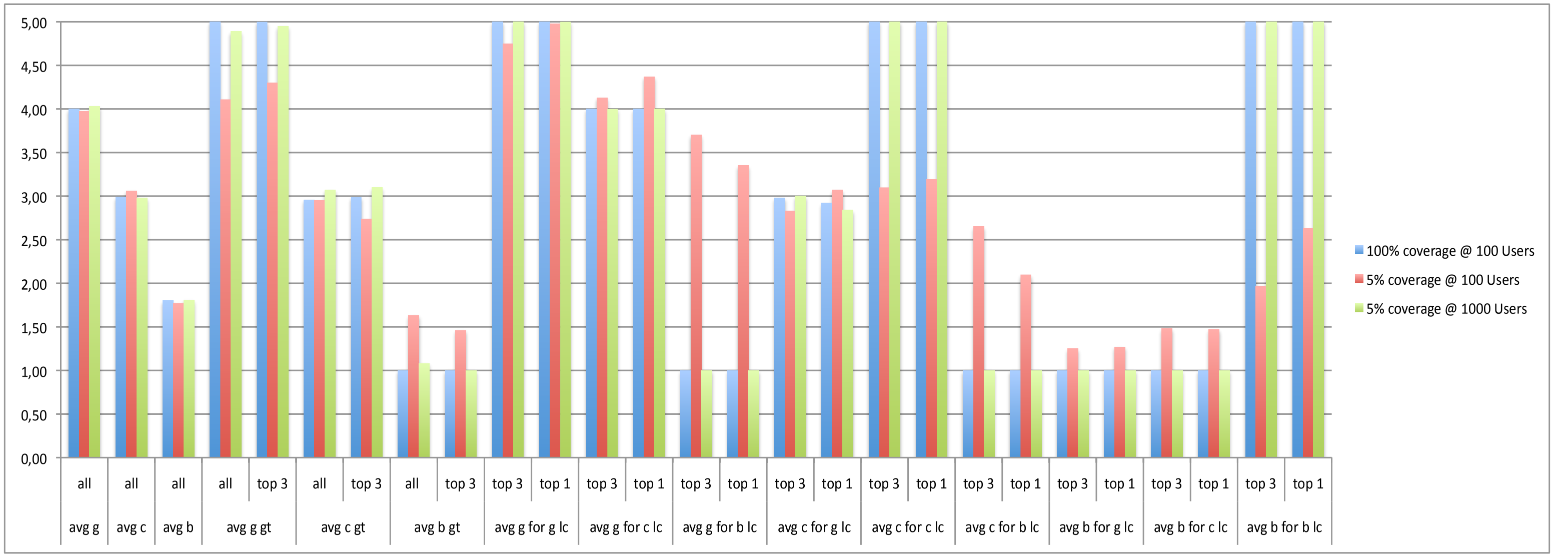
avg = average
g = good (ontology / user)
c = controversial (ontology / user)
b = bad (ontology / user)
all = based on all reviews
top 3 = based on top 3 reviews
top 1 = based on top review
gc = global trust (trust that is user independent)
lc = local trust (trust that is user dependent)
avg x for y lc = average ratings of x (e.g. good) ontologies for y (e.g. controversial) users based on localtrust

Fig. 9. Third Scenario with 20% Error

WOT. After all, the 5% coverage equals to two reviews and 5 trust statements in the 100 user setting. We have compared the influence of increasing the user-base while keeping the number of ontologies and level of coverage equal. The results can be found in Figure 10. We furthermore analyzed the setting with 1000 users in more detail, to find out how the derivations from the expected outcome can occur. As discussed in section 3.6, the theoretical lower boundary for a system that delivers expected results for our setup lies with 3 reviews per ontology (one for each group), and one trusted peer user for each ontology per user. That amounts to 3% coverage in reviews and 1% coverage in trust per ontology (in a 100 user setting). Since we have employed randomization for the simulation, the reviews and trust were not assigned optimally, but randomly based on the rules explained above. This could lead to settings where users are not connected to the WOT, and therefore for them, no local trust information can be used to compute the overall rating. In order to see how many users are disconnected for each ontology (since for all these users, in our experiment setup their trust results would contain global trust information), we ran an analysis on the setting. For the setup 100 users, 5% coverage and 0% error, the results can be found in Figure 11. The graphics displays, for how many ontologies the number of users disconnected from the WOT falls into the percentage of users displayed on the x-axis. As is evident from Figure 11, for several ontologies a larger percentage of users was not connected to the WOT, in some cases even more than 90%. Therefore it is not surprising that the average ratings based on local trust deviates towards the ratings based on global trust, since in the some cases, the ranking algorithm has to rely on global trust. In the rest of the cases, the correct review is retrieved based on local trust and is blended into the results. This is why the results deviate in direction of the expected results. Given a setting of 1000 users, 5% coverage and 0% error, there are almost no users disconnected from the WOT, as can be seen in Figure 12. If we remember that the minimal setting for achieving expected results is only partially dependent on the user size (the number of reviews required is independent from the number of users, but is based on the number of ontologies), we are not surprised to see improved results in the setting with more users (since the 5% here mean 10 times more reviews and trust statements than in the 100 user setting). As a conclusion of the comparison we note that the number of reviews and trust statement in total is not as important as having the right ones (the ones described in section 3.6). Given our experiment setup, by having a larger number of reviews and trust statements, we have achieved a better coverage and therefore better results.

3.6 Minimal Setting with Expected Behavior

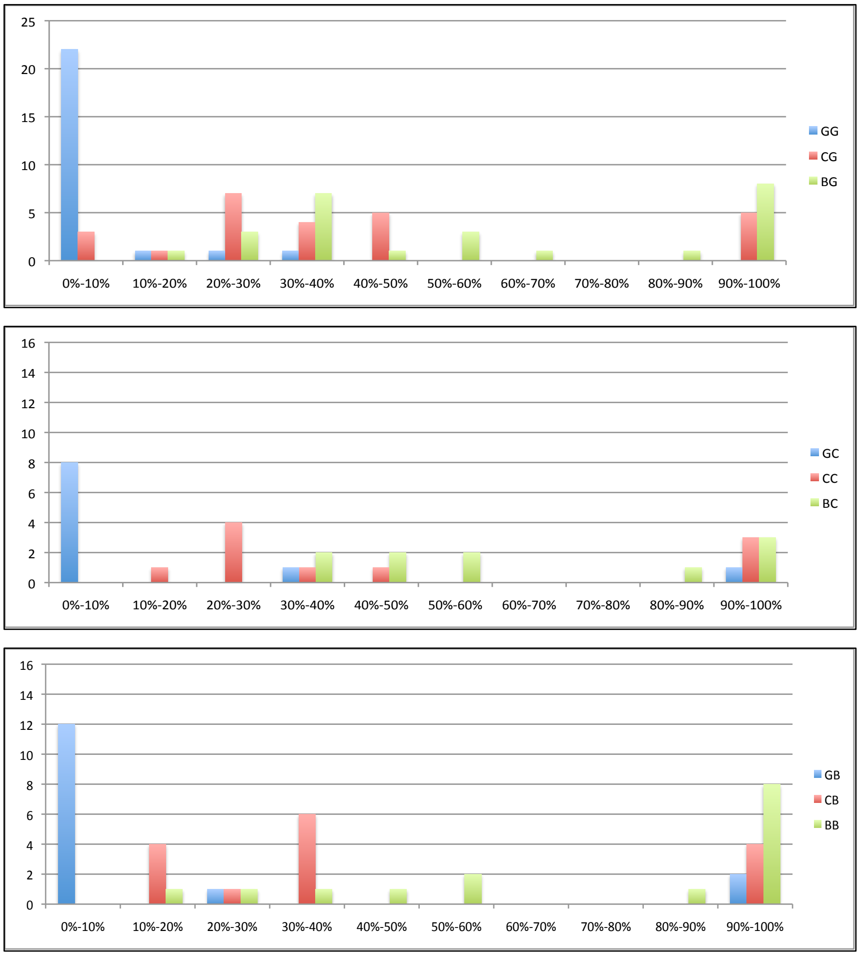
As discussed above, most of the problems with results deviating from expectations are due to users that are not connected to the WOT, which causes the system to fall back on global trust. So the more users are not connected to the WOT, the more the results resemble global trust results. A minimum setting yielding perfect results (as expected or achieved with 100% coverage) for all



Abbreviations:

avg = average
g = good (ontology / user)
c = controversial (ontology / user)
b = bad (ontology / user)
all = based on all reviews
top 3 = based on top 3 reviews
top 1 = based on top review
gc = global trust (trust that is user independent)
lc = local trust (trust that is user dependent)
avg x for y lc = average ratings of x (e.g. good) ontologies for y (e.g. controversial) users based on localtrust

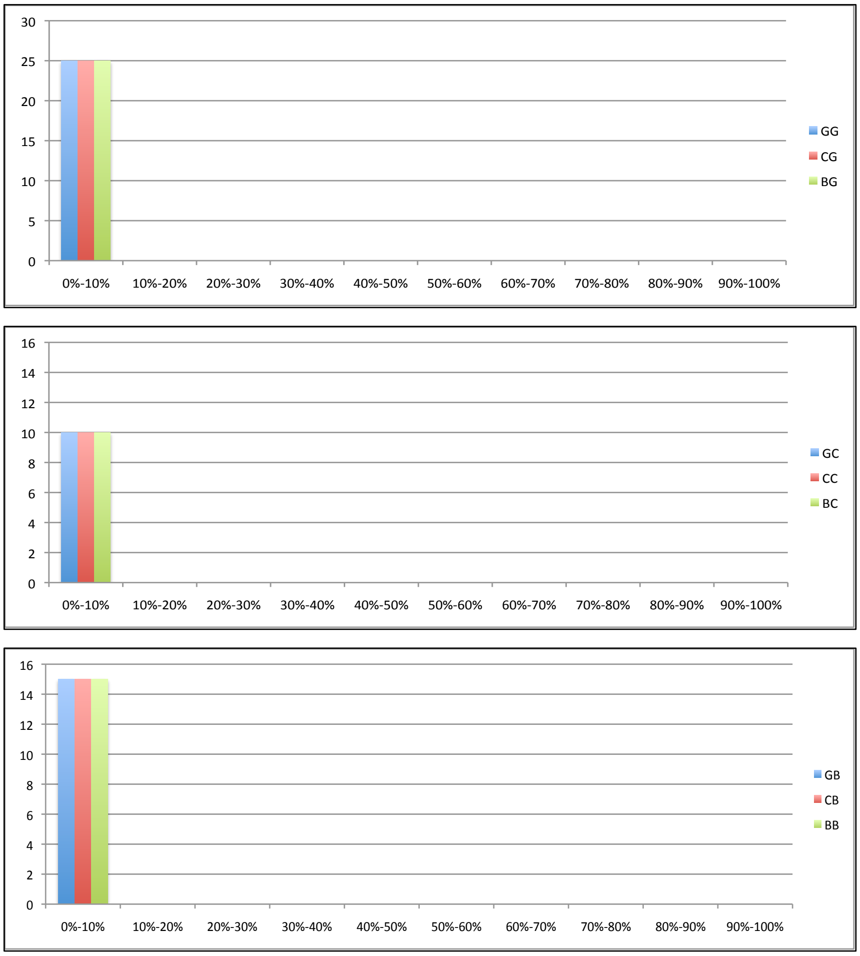
Fig. 10. Comparison of Results Scenario 1 0% Error



Abbreviations:

g = good (ontology / user)
c = controversial (ontology / user)
b = bad (ontology / user)
xy = x user on y ontologies (e.g. bc = results for bad users on controversial ontologies)

Fig. 11. For this graph, we have checked for each ontology in the investigated group, which percentage of the users in the group investigated has no local trust information for this ontology. The findings were sorted into 10% buckets. The computations were performed with 100 users, 5% coverage and 0% error.



Abbreviations:

g = good (ontology / user)
c = controversial (ontology / user)
b = bad (ontology / user)
xy = x user on y ontologies (e.g. bc = results for bad users on controversial ontologies)

Fig. 12. For this graph, we have checked for each ontology in the investigated group, which percentage of the users in the group investigated has no local trust information for this ontology. The findings were sorted into 10% buckets. The computations were performed with 1000 users, 5% coverage and 0% error.

users is achieved, when for each ontology at least one review exists from each user group (good, controversial, bad), and if each user trusts at least one review from a peer for each ontology in the system. This results in all users being connected to the WOT, and the reviewer from the peer group delivering the top-ranked result. Of course, if only one good review exists for each group, the results have to rely on the top review, not the top three reviews.

4 Conclusion

In the concluding section we want to revise what we have learned from the simulations, and how this can help to improve the accuracy of ranking in real world systems like cupboard.

4.1 Validity of Simulation Results for Real World Systems

We think that the simulation we ran on the one hand shows that the algorithms do work in a perfect scenario, but also show how derivations can influence the results. We purposely chose to employ randomness when running the less dense scenarios and those that contained errors, because randomness is the worst case scenario. In real life, it is more likely that users will not trust or distrust random users, but users whose reviews they read. This would lead to a scenario which is more specific and will thus be closer to what we described as the minimal setting with expected behavior.

4.2 Accuracy of Ranking Results for the Individual

All our scenarios were based on computing the average over all ratings for all users of a certain group. This has influenced the results to our disadvantage, because in fact even for those cases the algorithms work as expected given the data available. They provided the expected results for the users connected to the WOT and thus having local trust available. For the rest, the system had to rely on global trust, which deviated from the expected local results. The more users in a setting were not connected to the WOT, the closer the results were to global trust. In a real world scenario, a perfect coverage would not be required, because not all users would want to know about all other ontologies. Most times, users have a a clear idea what they need when searching for an ontology to reuse. In principle, a user will receive the correct score for each of the ontologies for which he has made a trust statement covering a review rating the ontology. So in the most ideal case, given that the user browsed the reviews for all properties of all ontologies and trusted one of the reviews per ontology–property connection, the rating results will be entirely based on local trust (i.e. the WOT), and will be as expected (see 3.1). Of course, this scenario is time-consuming and will likely not happen as such. So, the user must trust at least that many users on a meta-level (i.e. globally, per ontology or per property), that all ontology-property combinations are covered with reviews of trusted reviewers. If, for example, one reviewer reviews all the ontology-property connections the user is interested in, and the reviews can be trusted, a simple global trust statement from a user towards this one reviewer will be sufficient to provide the expected results. Especially in our data sparse setting, we can assume that in real world systems, the few existing trust statement would have been made by browsing ontologies of interest, and not randomly. Just by doing that, the local trust connectivity for the ontologies of interest would be higher than in our random scenario. We therefore argue that the results in real-world systems are better than those of our simulation using random behavior, simply because the users do not act randomly.

4.3 Significance of Error Settings

As mentioned before, when we tested for behavior of the ranking algorithms given erratic user behavior, we employed a random selection of users to trust and then had the users act in a certain percentage of the cases in contradiction to the rules defined (i.e. distrust instead or trust and vice versa). That means that a bad user would be trusted by a good user, but also that a good user might distrust another good user. While it was interesting to see how the results would be affected, we also argue that in reality, user behavior will not be that random and erratic. For any user not trying to act maliciously on purpose, there is no incentive to act in contrast to the best knowledge. And technically speaking, when a user enters a wrong statement into the system, the results of that behavior cannot be considered an error by the system, since the user is responsible for providing correct data to the system.

4.4 Attacks on the System

As could be shown in the bad user scenario, the local trust algorithms are resistant to most kinds of attacks, which we simulated with the bad uses group. As long as the malicious subgroup stays separate from the rest of the users in the WOT, no harm can be done. In case the bad users are only a small group in relation to the overall group, the damage done is low even in the cases where they are accidentally trusted. But even introducing large scales of bad users into the system (making it the predominant group) does not influence the local trust settings for the user connected to the WOT. The global trust can be affected (compare to the linkfarms [9] on the web trying to increase their Google pagerank), but it is only used for users who cannot be identified or are not connected to the WOT. Now one can argue that the bad users could try to gain the trust of normal users by acting like a good user in e.g. 90% of the cases, and then falsely review ontologies in the rest of the cases. This attack is not that easy, since we compute the WOT separately for each of the ontology-property combinations. That means that the bad user under normal circumstances would only be trusted in the cases where he is acting like a good user, and not in the other cases. The only way to draw an advantage would be by convincing a user to express a meta-trust-statement, also covering potentially malicious reviews. But even if that is possible, one could argue that the good the bad user has done in order to gain trust is overall better for the system than the bad caused by having a few malicious cases. In order to prevent being a victim of such a malicious reviewer, users should be very sensitive as to who they trust on a meta-level.

4.5 Lessons Learned

It is important that users make use of the trust function in order to receive personalized results. If they do not, they receive ratings based on global trust, which works fine as long as most users are good users and the user has needs similar to the mainstream. It is furthermore important that users know only to issue a meta-trust statement when they are certain the user is deserving it. It should not be their default action to meta-trust a user. Because reviewers can have different strengths and weaknesses, it is important to use a system allowing to assign trust fine-grained, like the TS-ORS (see scenario 3 vs. scenario 2). Even though the local trust algorithms are immune to attacks from confined subgroups, it is in the best interest of the operator of a real-world TS-ORS to shut out bad users and keep the system clean.

References

1. Lewen, H.: Implementation and Performance Evaluation of the Topic-Specific Trust Open Rating System. Technical report, Universität Karlsruhe (TH) (JUN 2009) `http://www.aifb.uni-karlsruhe.de/WBS/hle/paper/TR1.pdf`.
2. Lewen, H.: Facilitating Ontology Reuse with a Topic-Specific Trust Open Rating System. Technical report, Universität Karlsruhe (TH) (JUN 2009) `http://www.aifb.uni-karlsruhe.de/WBS/hle/paper/TR3.pdf`.
3. d'Aquin, M., Lewen, H.: Cupboard – a place to expose your ontologies to applications and the community. In: In Proceedings of the Demo Track of the 6th European Semantic Web Conference. (MAY 2009) to appear.
4. Lewen, H., Supekar, K., Noy, N., Musen, M.: Topic-Specific Trust and Open Rating Systems: An Approach for Ontology Evaluation. In: Proc. of the 4th International Workshop on Evaluation of Ontologies for the Web (EON2006) at the 15th International World Wide Web Conference (WWW 2006), Edinburgh, UK (MAY 2006)
5. Sabou, M., Angeletou, S., d'Aquin, M., Barrasa, J., Dellschaft, K., Gangemi, A., Lehmann, J., Lewen, H., Maynard, D., Mladenic, D., Nissim, M., Peters, W., Presutti, V., Villazon, B.: D2.2.1 methods for selection and integration of reusable components from formal or informal user specifications. NeOn Project Deliverable D2.2.1, The Open University (MAY 2007)
6. Guha, R.V., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In Feldman, S.I., Uretsky, M., Najork, M., Wills, C.E., eds.: WWW, ACM (2004) 403–412
7. Guha, R.: Open Rating Systems. Technical report, Stanford University, CA, USA (2003)
8. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1) (2004) 5–53
9. Wu, B., Davison, B.D.: Identifying link farm spam pages. In Ellis, A., Hagino, T., eds.: WWW (Special interest tracks and posters), ACM (2005) 820–829